

***FUNDAMENTAL OF COMPUTER
PROGRAMMING LAB***



SUBMITTED BY:

TANISHA BAGESHWARI

ECE(A)

Sr. no. 67

SUBMITTED TO:

Dr. MANISH SAILANI

FACULTY OF TECHNOLOGY, UNIVERSITY OF DELHI

PROGRAM 1

Ques.1) Matrix addition: write a program to perform addition of two 2-D arrays (matrix).

Code:

```
#include <stdio.h>
//additon of two 2-D arrays(matrices)
int main() {
    int r, c, a[100][100], b[100][100], sum[100][100], i, j;
    printf("Enter the number of rows (between 1 and 100): ");
    scanf("%d", &r);
    printf("Enter the number of columns (between 1 and 100): ");
    scanf("%d", &c);
    printf("\nEnter elements of 1st matrix:\n");
    for (i = 0; i < r; i++)
        for (j = 0; j < c; j++) {
            printf("Enter element a%d%d: ", i + 1, j + 1);
            scanf("%d", &a[i][j]);
        }
    printf("Enter elements of 2nd matrix:\n");
    for (i = 0; i < r; i++)
        for (j = 0; j < c; j++) {
            printf("Enter element b%d%d: ", i + 1, j + 1);
            scanf("%d", &b[i][j]);
        }
    // adding two matrices
    for (i = 0; i < r; i++)
        for (j = 0; j < c; j++) {
            sum[i][j] = a[i][j] + b[i][j];
        }
    // printing the result
    printf("\nSum of two matrices: \n");

    for (i = 0; i < r; i++)
        for (j = 0; j < c; j++) {
            printf("%d ", sum[i][j]);
            if (j == c - 1) {
                printf("\n\n");
            }
        }
    return 0;
}
```

Output:

```
Enter the number of rows (between 1 and 100): 2
Enter the number of columns (between 1 and 100): 3

Enter elements of 1st matrix:
Enter element a11: 1
Enter element a12: 4
Enter element a13: 1
Enter element a21: 3
Enter element a22: 4
Enter element a23: 5
Enter elements of 2nd matrix:
Enter element b11: 3
Enter element b12: 4
Enter element b13: 2
Enter element b21: 1
Enter element b22: 5
Enter element b23: 5

Sum of two matrices:
4 8 3

4 9 10
```

Ques.2)Matrix Multiplication: write a program to perform multiplication of two 2-D arrays.

Code:

```
#include <stdio.h>
// function to get matrix elements entered by the user
void getMatrixElements(int matrix[][10], int row, int column) {
    printf("\nEnter elements: \n");
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < column; j++) {
            printf("Enter a%d%d: ", i + 1, j + 1);
            scanf("%d", &matrix[i][j]);
        }
    }
}

// function to multiply two matrices
void multiplyMatrices(int first[][10],
                     int second[][10],
                     int result[][10],
                     int r1, int c1, int r2, int c2) {
    for (int i = 0; i < r1; i++) {
        for (int j = 0; j < c2; j++) {
            result[i][j] = 0;
        }
    }

    // Multiplying first and second matrices and storing it in result
    for (int i = 0; i < r1; i++) {
        for (int j = 0; j < c2; j++) {
            for (int k = 0; k < c1; k++) {
                result[i][j] += first[i][k] * second[k][j];
            }
        }
    }
}

// function to display the matrix
void display(int result[][10], int row, int column) {
    printf("\nOutput Matrix:\n");
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < column; j++) {
            printf("%d ", result[i][j]);
            if (j == column - 1)
                printf("\n");
        }
    }
}

int main() {
    int first[10][10], second[10][10], result[10][10], r1, c1, r2, c2;
    printf("Enter rows and column for the first matrix: ");
    scanf("%d %d", &r1, &c1);
    printf("Enter rows and column for the second matrix: ");
    scanf("%d %d", &r2, &c2);
    // Taking input until
    // 1st matrix columns is not equal to 2nd matrix row
    while (c1 != r2) {
        printf("Error! Enter rows and columns again.\n");
        printf("Enter rows and columns for the first matrix: ");
        scanf("%d %d", &r1, &c1);
    }
}
```

```

        printf("Enter rows and columns for the second matrix: ");
        scanf("%d%d", &r2, &c2);
    }

    // get elements of the first matrix
    getMatrixElements(first, r1, c1);
    // get elements of the second matrix
    getMatrixElements(second, r2, c2);
    // multiply two matrices.
    multiplyMatrices(first, second, result, r1, c1, r2, c2);
    // display the result
    display(result, r1, c2);
    return 0;
}

```

Output:

```

Enter rows and column for the first matrix: 2

2
Enter rows and column for the second matrix: 2

2

Enter elements:
Enter a11: 1
Enter a12: 3
Enter a21: 4
Enter a22: 6

Enter elements:
Enter a11: 3
Enter a12: 2
Enter a21: 8
Enter a22: 9

Output Matrix:
27 29
60 62

...Program finished with exit code 0
Press ENTER to exit console.

```

Program 3

Ques .3) string reverse without library function: write a program to reverse a string (array of char) without using string library functions .

Code:

```

#include <stdio.h>
#include <string.h>
//for reversing the string
int main()
{
    char str[100],revStr[100];
    int i,j;
    printf("Enter a string: ");
    scanf("%[^\n]s",str);

    j=0;
    for(i=(strlen(str)-1); i>=0;i--)
        revStr[j++]=str[i];

    revStr[j]='\0';
    printf("\nOriginal String is: %s",str);
    printf("\nReversed String is: %s",revStr);
    return 0;
}

```

Output:

```
Enter a string: Tanisha
Original String is: Tanisha
Reversed String is: ahsinaT
...Program finished with exit code 0
Press ENTER to exit console.
```

Program 4

Ques.4) : String Handling using Library Functions:

- Palindrome Check for a String: Write a program to check if a string is a palindrome or not.
- String Length Calculation Using Functions: Write a program to calculate the length of a string using a library functions/predefined operators.
- String Concatenation: Write a program to concatenate two strings using library functions.

Code : a.

```
#include <stdio.h>
#include <string.h>
//string is palindrome or not
int main()
{
    char str[] = { " " };
    printf("Enter a string: ");
    scanf("%s",str);
    int a = 0;
    int b = strlen(str) - 1;
    while ((a < b)) {
        if (str[b++] != str[a--]) {
            printf("%s is not a palindrome\n", str);
            return 0;
        }
    }
    printf("%s is a palindrome\n", str);
    return 0;
}
```

Output:

```
Enter a string: Tanisha
Tanisha is a palindrome
*** stack smashing detected ***: terminated
...Program finished with exit code 0
Press ENTER to exit console.
```

Code: b.

```
#include <stdio.h>
#include <string.h>
//calculate string length
int main()
{
    char Str[1000];
    int i;
    printf("Enter the String: ");
    scanf("%s", Str);
    for (i = 0; Str[i] != '\0'; i++);
    printf("Length of Str is %d", i);
    return 0;
}
```

output :

```
Enter the String: Mamta
Length of Str is 5

...Program finished with exit code 0
Press ENTER to exit console.
```

Code : c.

```
#include <stdio.h>
//string concatenation
int main() {
    char s1[100] = " ", s2[] = " ";
    printf("Enter s1: ");
    scanf("%s", s1);
    printf("Enter s2: ");
    scanf("%s", s2);
    int length, j;
    length = 0;
    while (s1[length] != '\0') {
        ++length;
    }
    for (j = 0; s2[j] != '\0'; ++j, ++length) {
        s1[length] = s2[j];
    }
    s1[length] = '\0';
    printf("After concatenation: ");
    puts(s1);
    return 0;
}
```

Output:

```
Enter s1: Hello
Enter s2: Maa

...Program finished with exit code 0
Press ENTER to exit console.
```

Program 5

Ques .5) Compute nCr using Functions: Write a function to calculate nCr , where n and r are given by the user.

Code:

```
#include <stdio.h>
int fact(int);
int main()
{
    int n,r,ncr;
    printf("Enter a number n\n");
    scanf("%d",&n);
    printf("Enter a number r\n");
    scanf("%d",&r);
    ncr=fact(n)/(fact(r)*fact(n-r));
    printf("Value of %dC%d = %d\n",n,r,ncr);
}
int fact(int n)
{
    int i,f=1;
    for(i=1;i<=n;i++)
    {
        f=f*i;
    }
    return f;
}
```

Output:

```
Enter a number n
4
Enter a number r
2
Value of 4C2 = 6

...Program finished with exit code 0
Press ENTER to exit console.
```

Program 6:

Ques.6) : Recursive Function for Fibonacci Series: Write a program to generate the Fibonacci series up to n terms using recursion.

Code:

```
#include<stdio.h>
//fibonacci series
int main(){
    int first=0, second=1, i, n, sum=0;
    printf("Enter the number of terms: ");
    scanf("%d",&n);
    printf("Fibonacci Series:");
    for(i=0 ; i<n ; i++)
    {
        if(i <= 1)
        {
            sum=i;
        }
        else
        {
            sum=first + second;
            first=second;
            second=sum;
        }
        printf(" %d",sum);
    }
    return 0;
}
```

Output :

```
Enter the number of terms: 8
Fibonacci Series: 0 1 1 2 3 5 8 13

...Program finished with exit code 0
Press ENTER to exit console.
```

Program 7

Ques.7) File Handling Operations –

- File Write Operation: Write a program to write data into a file.
- File Read Operation: Write a program to read data from a file.
- File Append Operation: Write a program to append data to a file.

Code: a.

```
#include <stdio.h>

int main() {
    FILE *file;
    char data[100];
    // Open a file for writing
    file = fopen("example.txt", "w");
    if (file == NULL) {
        printf("Error opening the file.\n");
        return 1;
    }
    // Get data from the user
    printf("Enter data to write to the file:\n");
    fgets(data, sizeof(data), stdin);
    // Write data to the file
    fprintf(file, "%s", data);
    // Close the file
    fclose(file);
    printf("Data written to the file successfully.\n");

    return 0;
}
```

Output:

```
Enter data to write to the file:
Tanisha
Data written to the file successfully.

...Program finished with exit code 0
Press ENTER to exit console.
```

Code: b.

```
#include <stdio.h>

int main() {
    FILE *file;
    char data[100];
    // Open a file for reading
    file = fopen("example.txt", "r");
    if (file == NULL) {
        printf("Error opening the file.\n");
        return 1;
    }
    // Read data from the file
    fscanf(file, "%[^\n]", data);
    // Close the file
    fclose(file);
    // Display the read data
    printf("Data read from the file:\n%s\n", data);

    return 0;
}
```


Output:

```
Error opening the file.

...Program finished with exit code 1
Press ENTER to exit console.
```

Code: c.

```
#include <stdio.h>

int main() {
    FILE *file;
    char data[100];
    // Open a file for appending
    file = fopen("example.txt", "a");
    if (file == NULL) {
        printf("Error opening the file.\n");
        return 1;
    }

    // Get data from the user
    printf("Enter data to append to the file:\n");
    fgets(data, sizeof(data), stdin);

    // Append data to the file
    fprintf(file, "%s", data);

    // Close the file
    fclose(file);
    printf("Data appended to the file successfully.\n");
    return 0;
}
```

Output:

```
Enter data to append to the file:
Tanisha
Data appended to the file successfully.

...Program finished with exit code 0
Press ENTER to exit console.
```

Program 8

Ques.8) Multi-level Menu using Switch Case: Write a program to create a multilevel menu using switch cases that perform various basic mathematical operations (Add, Subtract, Divide, Multiply, Exponent).

Code:

```
#include<stdio.h>
#include<math.h>
int main()
{
    int a,b;
    int op;
    printf(" 1.Addition\n 2.Subtraction\n 3.Multiplication\n 4.Division\n 5.Exponent\n");
    printf("Enter the values of a & b: ");
    scanf("%d %d",&a,&b);
    printf("Enter your Choice : ");
    scanf("%d",&op);
    switch(op)
    {
        case 1 :
            printf("Sum of %d and %d is : %d",a,b,a+b);
            break;
        case 2 :
            printf("Difference of %d and %d is : %d",a,b,a-b);
            break;
        case 3 :
            printf("Multiplication of %d and %d is : %d",a,b,a*b);
            break;
        case 4 :
            printf("Division of Two Numbers is %d : ",a/b);
            break;
    }
```

```

case 5 :
    printf("Exponent :%2lf\n ",pow(a,b));
    break;
default :|
    printf(" Enter Your Correct Choice.");
    break;
}
return 0;
}

```

Output:

```

1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Exponent
Enter the values of a & b: 2
4
Enter your Choice : 5
Exponent :16.000000

...Program finished with exit code 0
Press ENTER to exit console.

```

Program 9

Ques.9) : Linear Search in Array: Write a program to implement linear search in a 1-D array.

Code:

```

#include <stdio.h>
int linearSearch(int array[], int size, int key) {
    for (int i = 0; i < size; i++) {
        if (array[i] == key) {
            return i;
        }
    }
    return -1;
}

int main() {
    int array[100], size, key;
    printf("Enter the size of the array: ");
    scanf("%d", &size);
    printf("Enter the elements of the array:\n");
    for (int i = 0; i < size; i++) {
        printf("Enter element %d: ", i + 1);
        scanf("%d", &array[i]);
    }

    printf("Enter the key to search: ");
    scanf("%d", &key);
    int result = linearSearch(array, size, key);
    if (result != -1) {
        printf("Key found at index: %d\n", result);
    } else {
        printf("Key not found in the array.\n");
    }
    return 0;
}

```

Output:

```

Enter the size of the array: 5
Enter the elements of the array:
Enter element 1: 3
Enter element 2: 4
Enter element 3: 2
Enter element 4: 4
Enter element 5: 5
Enter the key to search: 3
Key found at index: 0

...Program finished with exit code 0
Press ENTER to exit console.

```

Program 10

Ques.10) Swap Two Numbers Using Pointers: Write a program to swap values of two variables using pointers.

Code:

```

#include <stdio.h>

void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

int main() {
    int num1, num2;
    printf("Enter value for num1: ");
    scanf("%d", &num1);
    printf("Enter value for num2: ");
    scanf("%d", &num2);
    printf("Before swapping: num1 = %d, num2 = %d\n", num1, num2);
    swap(&num1, &num2);
    printf("After swapping: num1 = %d, num2 = %d\n", num1, num2);
    return 0;
}

```

Output:

```

Enter value for num1: 2
Enter value for num2: 5
Before swapping: num1 = 2, num2 = 5
After swapping: num1 = 5, num2 = 2

...Program finished with exit code 0
Press ENTER to exit console.

```

Program 11

Ques.11) Pointer Arithmetic: Demonstrating basic pointer arithmetic operations (addition, subtraction). Write a program to illustrate the following basic pointer arithmetic operations.

- Increment a pointer
- Decrement a Pointer
- Add an integer to a pointer
- Subtract an integer from a pointer
- Subtract two pointers of the same type

code:

```
#include <stdio.h>

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int*ptr = arr;
    printf("Original value at *ptr: %d\n", *ptr);
    ptr++;
    printf("After incrementing, value at *ptr: %d\n", *ptr);
    ptr--;
    printf("After decrementing, value at *ptr: %d\n", *ptr);
    ptr = arr;
    printf("Original value at *ptr: %d\n", *ptr);
    ptr = ptr + 2;
    printf("After adding 2, value at *ptr: %d\n", *ptr);
    ptr = ptr - 1;
    printf("After subtracting 1, value at *ptr: %d\n", *ptr);
    int*ptr2 = arr + 3;
    printf("Value at *ptr2: %d\n", *ptr2);
    printf("Difference between ptr and ptr2: %ld\n", ptr2 - ptr);
    return 0;
}
```

Output:

```
Original value at *ptr: 1
After incrementing, value at *ptr: 2
After decrementing, value at *ptr: 1
Original value at *ptr: 1
After adding 2, value at *ptr: 3
After subtracting 1, value at *ptr: 2
Value at *ptr2: 4
Difference between ptr and ptr2: 2

...Program finished with exit code 0
Press ENTER to exit console.
```

Program 12

Ques.12) : Recursive Function for Factorial: Write a program to calculate the factorial of a number using recursion.

Code:

```
#include <stdio.h>
// Recursive function to calculate factorial
unsigned long long int factorial(int n) {
    if (n == 0 || n == 1) {
        return 1;
    } else {
        return n * factorial(n - 1);
    }
}

int main() {
    int num;
    printf("Enter a non-negative integer: ");
    scanf("%d", &num);
    if (num < 0) {
        printf("Please enter a non-negative integer.\n");
        return 1;
    }
    unsigned long long result = factorial(num);
    printf("Factorial of %d = %llu\n", num, result);
    return 0;
}
```

Output:

```
Enter a non-negative integer: 5
Factorial of 5 = 120

...Program finished with exit code 0
Press ENTER to exit console.
```

Program 13

Ques.13) : Call by Reference Using Pointers: Write a program to demonstrate call by reference using pointers in functions.

Code:

```
#include <stdio.h>
void swap(int *a, int *b);
int main() {

    int num1, num2;
    printf("Enter the value for num1: ");
    scanf("%d", &num1);
    printf("Enter the value for num2: ");
    scanf("%d", &num2);
    printf("Before swapping: num1 = %d, num2 = %d\n", num1, num2);

    // Call the swap function with call by reference
    swap(&num1, &num2);
    printf("After swapping: num1 = %d, num2 = %d\n", num1, num2);
    return 0;
}

// Function to swap values using pointers (call by reference)
void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}
```

Output:

```
Enter the value for num1: 4
Enter the value for num2: 5
Before swapping: num1 = 4, num2 = 5
After swapping: num1 = 5, num2 = 4

...Program finished with exit code 0
Press ENTER to exit console.
```

Program 14

Ques.14) Dynamic Memory Allocation for 1-D Array: Write a program to allocate memory dynamically for a 1-D array using malloc.

Code:

```
#include<stdio.h>
#include<stdlib.h>
int main(){
    int*arr;
    int size;
    printf("Enter the size of the array: ");
    scanf("%d",&size);
    arr=(int*)malloc(sizeof(int));
    if(arr==NULL){
        printf("Memory allocation failed.Existing program.\n");
        return 1;
    }

    printf("Enter %d elements for the array:\n",size);
    for(int i=0;i<size;i++){
        printf("Enter element %d: ",i+1);
        scanf("%d",&arr[i]);
        printf("Elements of dynamically allocated array:\n");
        for(int i;i<size;i++){
            printf("%d",arr[i]);
        }
        printf("\n");
        free(arr);
        return 0;
    }
}
```

Output:

```

Enter the size of the array: 4
Enter 4 elements for the array:
Enter element 1: 3
Enter element 2: 2
Enter element 3: 4
Enter element 4: 6
Elements of dynamically allocated array:
3246

...Program finished with exit code 0
Press ENTER to exit console.

```

Program 15

Ques.15) Sum of Diagonal Elements in a Matrix: Write a program to calculate the sum diagonal elements of a 2-D square matrix using a function.

Code:

```

#include <stdio.h>
// Function to calculate the sum of diagonal elements in a square matrix
int sumOfDiagonal(int matrix[10][10], int size) {
    int sum = 0;
    for (int i = 0; i < size; i++) {
        sum += matrix[i][i];
    }
    return sum;
}

int main() {
    int matrix[10][10];
    int size;
    printf("Enter the size of the square matrix: ");
    scanf("%d", &size);
    printf("Enter elements for the square matrix:\n");
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            printf("Enter element at position (%d, %d): ", i + 1, j + 1);
            scanf("%d", &matrix[i][j]);
        }
    }
}

```

```

    printf("\nThe entered matrix is:\n");
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            printf("%d\t", matrix[i][j]);
        }
        printf("\n");
    }
    int diagonalSum = sumOfDiagonal(matrix, size);
    printf("\nThe sum of diagonal elements is: %d\n", diagonalSum);
    return 0;
}

```

Output:

```

Enter the size of the array: 4
Enter 4 elements for the array:
Enter element 1: 3
Enter element 2: 2
Enter element 3: 4
Enter element 4: 6
Elements of dynamically allocated array:
3246

...Program finished with exit code 0
Press ENTER to exit console.

```